

**AMENDMENTS TO THE CLAIMS**

1. (Currently amended): A logically partitioned data processing system, comprising:  
a plurality of operating systems, each assigned to a separate one of a plurality of logical partitions;

a hypervisor for creating and enforcing separation between each of the plurality of logical partitions; wherein

the hypervisor includes a plurality of ~~functions~~ selectable function sets, each function set including a list of hypervisor function calls that are provided by the hypervisor ~~functions~~ which may be called by any one of the plurality of operating systems to get said hypervisor to perform tasks for the calling one of the plurality of operating systems a task while maintaining separation between each of the plurality of logical partitions,

said hypervisor receiving a selection of one of said plurality of function sets;

said selected one of said plurality of function sets being enabled;

the hypervisor informs each of the plurality of operating systems of an enabled function set, ~~wherein functions~~ function calls included identified within the enabled function set being ~~being~~ [[are]] enabled for use by each of the plurality of operating systems and ~~functions~~ function calls not included identified within the enabled function set being ~~being~~ [[are]] disabled for use by each of the plurality of operating systems.

2. (Original): The logically partitioned data processing system as recited in claim 1, wherein the enabled function set from the plurality of function sets may be changed such that a different one of the plurality of function sets becomes the enabled function set.

3. (Currently amended): The logically partitioned data processing system as recited in claim 1, wherein additional function sets may be added to the plurality of function sets as additional function calls ~~functions~~ are added to the plurality of function calls ~~functions~~ provided by the hypervisor.

4. (Original): The logically partitioned data processing system as recited in claim 1, wherein the hypervisor is implemented as firmware.

5. (Currently amended): The logically partitioned data processing system as recited in claim 1, wherein each of the plurality of function sets comprises a different group of the plurality of function calls ~~functions~~.

6. (Currently amended): The logically partitioned data processing system as recited in claim 1, wherein optional function calls ~~functions~~ are omitted from at least one of the plurality of function sets.

7. (Currently amended): A method of identifying hypervisor function calls that are currently available for execution by configuring a set of services provided by a hypervisor in ~~in~~ [[to]] a logically partitioned data processing system, the method comprising:

assigning each one of a plurality of operating systems to a separate one of a plurality of logical partitions;

creating and enforcing, by said hypervisor, separation between each of the plurality of logical partitions;

providing, by said hypervisor, a plurality of selectable function sets, each function set including a list of function calls that are provided by the hypervisor that can be called by any one of the plurality of operating systems to get said hypervisor to perform tasks for the calling one of the plurality of operating systems while maintaining separation between each of the plurality of logical partitions;

receiving, by said hypervisor, a selection of one of said plurality of function sets;

enabling said selected one of said plurality of function sets;

informing, by said hypervisor, each one of the plurality of operating systems of an enabled function set, function calls included within the enabled function set being enabled for use by each one of the plurality of operating systems and function calls not included within the enabled function set being disabled for use by each one of the

plurality of operating systems, wherein hypervisor function calls that are currently available for execution are identified.

~~presenting a user with a set of service options, wherein the set of service options correspond to services performed by the hypervisor for each of multiple operating systems within the logically partitioned data processing system such that processes performed by one of the multiple operating systems do not interfere with processes performed by others of the multiple operating systems;~~

~~responsive to selection of a particular service option, storing the selected service option and presenting the service option to an operating system image as the operating system image is initialized.~~

8. (Currently amended): The method as recited in claim 7, further comprising:  
responsive to loading a new version of the hypervisor, ~~wherein~~ the new version of the hypervisor ~~contains~~ containing additional function calls ~~services~~, reporting the additional function calls ~~services~~ to each operating system upon re-initialization.

9. (Original): The method as recited in claim 7, wherein the operating system image is initialized by booting.

10. (Original): The method as recited in claim 8, wherein the re-initialization of each operating system is performed by a reboot.

11. (Original): The method as recited in claim 7, wherein the hypervisor is implemented as firmware.

12. (Canceled)

13. (Currently amended): A computer program product in a computer readable media for use in a data processing system for identifying hypervisor function calls that are currently available for execution by ~~configuring a set of services provided by a~~

hypervisor ~~[[to]]~~ in a logically partitioned data processing system, the computer program product comprising:

instructions for assigning each one of a plurality of operating systems to a separate one of a plurality of logical partitions;

instructions for creating and enforcing, by said hypervisor, separation between each of the plurality of logical partitions;

instructions for providing, by said hypervisor, a plurality of selectable function sets, each function set including a list of function calls that are provided by the hypervisor that can be called by any one of the plurality of operating systems to get said hypervisor to perform tasks for the calling one of the plurality of operating systems while maintaining separation between each of the plurality of logical partitions;

instructions for receiving, within said hypervisor, a selection of one of said plurality of function sets;

instructions for enabling said selected one of said plurality of function sets;

instructions for informing, by said hypervisor, each one of the plurality of operating systems of an enabled function set, function calls included within the enabled function set being enabled for use by each one of the plurality of operating systems and function calls not included within the enabled function set being disabled for use by each one of the operating systems, wherein hypervisor function calls that are currently available for execution are identified.

~~first instructions for presenting a user with a set of service options, wherein the set of service options correspond to services performed by the hypervisor for each of multiple operating systems within the logically partitioned data processing system such that processes performed by one of the multiple operating systems do not interfere with processes performed by others of the multiple operating systems; and~~

~~second instructions, responsive to selection of a particular service option, for storing the selected service option and presenting the service option to an operating system image as the operating system image is initialized.~~

14. (Currently amended): The computer program product as recited in claim 13, further comprising:

[[third]] instructions, responsive to loading a new version of the hypervisor, wherein the new version of the hypervisor containing ~~contains~~ additional function calls services, for reporting the additional function calls services to each operating system upon re-initialization.

15. (Original): The computer program product as recited in claim 13, wherein the operating system image is initialized by booting.

16. (Original): The computer program product as recited in claim 14, wherein the re-initialization of each operating system is performed by a reboot.

17. (Original): The computer program product as recited in claim 13, wherein the hypervisor is implemented as firmware.

18. (Canceled)

19. (Currently amended): A system for identifying hypervisor function calls that are currently available for execution by configuring a set of services provided by a hypervisor ~~[[to]]~~ in a logically partitioned data processing system, the system comprising:  
means for assigning each one of a plurality of operating systems to a separate one of a plurality of logical partitions;

means for creating and enforcing, by said hypervisor, separation between each of the plurality of logical partitions;

means for providing, by said hypervisor, a plurality of selectable function sets, each function set including a list of function calls that are provided by the hypervisor that can be called by any one of the plurality of operating systems to perform tasks for the calling one of the plurality of operating systems while maintaining separation between each of the plurality of logical partitions;

means for receiving a selection of one of said plurality of function sets;

means for enabling said selected one of said plurality of function sets;

means for informing, by said hypervisor, each one of the plurality of operating systems of an enabled function set, function calls included within the enabled function set being enabled for use by each one of the plurality of operating systems and function calls not included within the enabled function set being disabled for use by each one of the operating systems, wherein hypervisor function calls that are currently available for execution are identified.

~~first means for presenting a user with a set of service options, wherein the set of service options correspond to services performed by the hypervisor for each of multiple operating systems within the logically partitioned data processing system such that processes performed by one of the multiple operating systems do not interfere with processes performed by others of the multiple operating systems; and~~

~~second means, responsive to selection of a particular service option, for storing the selected service option and presenting the service option to an operating system image as the operating system image is initialized.~~

20. (Currently amended): The system as recited in claim 19, further comprising:

[[third]] means, responsive to loading a new version of the hypervisor, wherein the new version of the hypervisor containing ~~contains~~ additional function calls ~~services~~, for reporting the additional function calls ~~services~~ to each operating system upon re-initialization.

21. (Original): The system as recited in claim 19, wherein the operating system image is initialized by booting.

22. (Original): The system as recited in claim 20, wherein the re-initialization of each operating system is performed by a reboot.

23. (Original): The system as recited in claim 19, wherein the hypervisor is implemented as firmware.

24. (Canceled)

25. (New): The method according to claim 7, further comprising:  
receiving a selection of said one of said plurality of function sets from a platform  
included within said data processing system.
26. (New): The method according to claim 7, further comprising:  
during booting of each one of said plurality of operating systems, passing a  
parameter to each one of said plurality of operating systems that identifies said selected  
one of said plurality of function sets.
27. (New): The method according to claim 7, further comprising:  
receiving said selection of said one of said plurality of function sets from a user.